arXiv:2302.04460v2 [cs.CL] 1 Jun 2023

Bag 语言模型训练数据提取技巧

Weichen Yu [*1] **Tian Yu Pang** [2] **Qian Liu** [2] **Chao Du** [2] **Bingyi Kang** [2] **Yan Huang** [1] **Min Lin** [2] **Shuicheng Yan** [2]

摘要

随着语言模型的发展，隐私保护正受到越来越多的关注。因此，训练数据提取具有重要意义，因为它可以作为评估隐私泄露的潜在工具。然而，由于该任务难度较高，现有的大多数方法仍停留在概念验证阶段，效果也不够理想。本文利用一个公开数据集，研究并评测了用于改进训练数据提取的技巧。由于现有的大多数提取方法都采用“先生成后排序”的流程，即先生成文本候选项作为潜在的训练数据，再根据特定标准对其进行排序，因此我们的研究重点关注文本生成（例如采样策略）和文本排序（例如词元级标准）两方面的技巧。实验结果表明，一些此前被忽视的技巧对于训练数据提取的成功至关重要。根据 GPT-Neo 1.3B 的评测结果，我们提出的技巧在大多数情况下都以较大幅度优于基线，为未来研究提供了更强的基线。代码可从 https://github.com/weichen-yu/LM-Extraction获取。

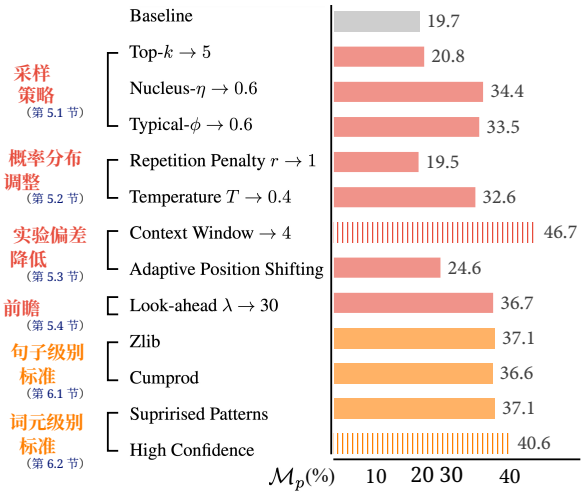

图 *1.* 本文所探索技巧组合的概览，以及精确率（$\mathcal{M}_P$）评估。 粉色 柱表示改进的后缀生成方法，橙色 柱表示改进的后缀排序方法。虚线柱 表示每个类别中的最佳方法。

1. 引言

语言模型（LM）的最新进展Kenton& Toutanova，2019年；Brown 等人，2020年）。然而，已有研究表明，由于Kenton & Toutanova，2019年；Carlini 等人，2019年；Feldman，2020年；Brown 等人，2020年）的记忆效应，可以从语言模型中提取训练数据。这些训练数据可能包含姓名、电子邮件地址、电话号码和实际地址等敏感信息，从而导致隐私泄露，阻碍语言模型的广泛应用（Carlini 等人，2021年；2022年；Lehman 等人，2021年）。

鉴于隐私安全一直是公众关注的重要问题，当前的一个关键课题是开发用于评估隐私泄露的高效方法。因此，我们的研究重点是针对语言模型训练数据提取这一对抗性任务；这是一个相对较新的研究领域（Carlini 等人，2021年；Lehman 等人，2021年）。现有的提取方法已经成功提取出一些记录，但在某些情况下，这些方法甚至不如根据先验分数直接选择最受欢迎的实体有效。此外，成功的数据提取需要较高的生成比例，即需要生成并排序大量候选项，才能识别出一个成功实例。这些次优结果表明，尽管先前研究已经证明训练数据提取的可行性并开发了开创性方法，但这项任务仍然相对较新，且有大量问题亟待解决。

在本研究中，我们旨在开发高效提取训练数据的技术。我们遵循近期训练数据提取挑战赛的标准，[1] 该挑战赛采用

[*]在 Sea AI Lab 实习期间完成。 [1]中国科学院自动化研究所。 [2]Sea AI Lab。通信作者：Tianyu Pang <tianyupang@sea.com>、Qian Liu< liuqian@sea.com>、Yan Huang <yhuang@nlpr.ia.ac.cn>。

《 40th 国际机器学习会议》，美国夏威夷州檀香山。PMLR 202, 2023。版权所有 2023，归作者所有。

[1]网站链接 训练数据提取挑战赛

一个拥有 13 亿参数的 GPT-Neo 模型（Black 等人，2021年）进行定向提取1-情景式记忆数据。定向提取指的是提供数据前缀的场景，例如“Yu 的电话号码是”，攻击者试图恢复后缀“12345”。根据Carlini 等人（2021年），κ-情景式记忆被定义为语言模型记住在训练材料中出现κ 次的字符串的能力。定向的1-情景式提取带来的风险更大，且比非定向和 κ-情景式（针对 $\kappa > 1$）设置更具挑战性。

通过消融研究，我们评估了自然语言处理（NLP）中的多种简单技巧，并实证评估了它们对成功提取率的影响，如图 1 所示。我们的实证分析表明，提取效果可能对实验设置十分敏感。在设置恰当的情况下，结果显示，若干以往被忽视的技巧有助于显著提升训练数据提取效果。根据 GPT-Neo 1.3B 的评估结果，我们提出的技巧在大多数情况下都大幅优于基线，为未来研究提供了强得多的基线。尽管如此，使用多种训练数据提取技巧并不一定能提升性能；在某些情况下，这些技巧甚至会相互不兼容，导致精确率下降。这些发现表明，审慎地选择和组合这些技巧对于实现最佳性能至关重要。

2. 相关工作

我们简要介绍训练数据提取、成员推断攻击以及其他基于记忆的攻击。

2.1. 训练数据提取

从预训练语言模型中提取训练数据，也称为语言模型数据提取，是一种恢复用于训练模型的示例的方法。尽管这是一项相对较新的任务，但许多基础技术和分析方法早已提出，包括成员推断（Shokri 等，2017年）以及利用网络记忆发起攻击（Thomas 等人，2020年；Leino 和 Fredrikson，2020年）。

Carlini 等人（2021年）率先定义了模型知识提取和 κ 情景式记忆的概念，并提出了有前景的数据提取训练策略。记忆的理论属性以及模型提取在敏感领域中的应用（例如临床笔记分析），一直是该领域后续研究的重点。

最近，Kandpal 等人(2022年) 证明，在语言模型中，数据提取之所以经常有效，往往是因为常用的网络抓取训练集中存在重复。利用非确定性，Jagielski 等人(2022年) 解释了如何遗忘记忆样本。Carlini 等人(2022年) 分析了影响训练数据记忆的三个因素。对于自然数据分布，Feldman(2020年) 表明，要实现近乎最优的性能，必须进行标签记忆。在模型提取的应用中，Lehman 等人 (2021年) 指出，预训练的 BERT 若在临床笔记上训练，会带来敏感数据泄露风险，尤其是在数据重复率较高或存在“注释膨胀”时（Liu 等人，2022年）。Jayaraman 等人(2022年) 提出了一种主动提取攻击，利用规范模式和差分隐私来防御模式提取攻击。

2.2. 成员推断攻击

成员推断攻击（MIA）（Shokri 等，2017年）是数据保护领域另一项与训练数据提取密切相关的对抗性任务。它旨在通过黑盒访问模型，确定某条特定记录是否存在于模型的训练数据集中。MIA 已被证明在许多机器学习任务中有效，包括分类（Sablayrolles 等人，2019年；Choquette-Choo 等人，2021年；Rezaei 和 Liu，2021年）和生成（Hayes 等人，2019年；Hilprecht 等人，2019年）。

MIA 所采用的方法分为两类：基于分类器的方法和基于指标的方法（Hu 等，2022）。基于分类器的方法通过训练二元分类器来识别成员与非成员之间的复杂关系，其中影子训练是一种常用技术（Shokri 等，2017年；He 等，2020年；Wang 等，2021年）。另一方面，基于指标的方法先计算模型预测向量上的指标，再据此进行成员推断（Yeom 等，2018年；Salem 等，2018年；Sablayrolles 等，2019；Song 和 Mittal，2021；Choquette-Choo 等，2021）。目前已有多种防御方法被提出，用于缓解这一漏洞，包括基于差分隐私（Leino 和 Fredrikson，2020年；Naseri 等，2020年；Choquette-Choo 等，2021）、数据剪枝（Wang 等，2021年）、数据增强（Kaya 和 Dumitras，2021）以及因果推断（Tople 等，2020）的方法。

2.3. 其他基于记忆的攻击

研究发现，大型预训练模型容易记住训练数据中的信息，这可能导致多种攻击。除了训练数据提取和成员推断攻击之外，还有其他针对这些模型的基于记忆的攻击。

模型提取攻击及其相应的保护方法（Tram`er 等人，2016年；Juuti 等人，2019年；Gong 等人，2020年；Wu 等人，2022年）关注的是复制给定模型功能的问题。在这类攻击中，攻击者试图构建第二个模型，使其预测性能与原始黑盒模型相近。

属性推断攻击的目标是从模型中提取特定的个人属性，例如地点、职业和兴趣（Fredrikson 等人，2015年；Gong 和 Liu，2016 年；Ganju 等人，2018 年；Parisot 等人，2021年）。属性推断攻击的目标是提取模型提供者可能无意分享的训练数据属性，例如生成数据的环境，或属于某个特定类别的数据比例。训练数据提取攻击与属性／属性推断攻击的主要区别在于，属性推断攻击不需要事先知道要提取的属性，而训练数据提取攻击则要求生成的信息在句子级别与训练数据完全一致，因此难度更大、危害也更高。

3. 预备知识

我们回顾本研究采用的基本设置。这些设置主要遵循训练数据提取挑战赛的指南。随后，我们定义威胁模型和评估指标。

3.1. 基本设置

数据集。本研究使用的数据集是从 Pile 的训练数据集中抽取的 20,000 个样例子集（Gao 等人，2020年）。每个样例由一个包含 50 个词元的前缀和一个包含 50 个词元的后缀组成。攻击者的任务是根据前缀预测后缀。该数据集中的所有 100 词元长的句子在训练集中都只出现一次。在本研究中，我们将数据集划分为包含 19,000 个样本的训练集和包含 1,000 个样本的测试集。

语言模型。我们采用了 HuggingFace Transformers 上实现的 GPT-Neo 1.3B 模型（Wolf 等人，2020年）。该 Transformer 模型基于 EleutherAI 对 GPT-3 架构的复现而设计（Brown 等人，2020年），并在 Pile 数据集上训练。GPT-Neo 是一个自回归语言模型 f_θ，其参数化为 θ，可通过链式法则 x_0生成词元序列 x_1，$\cdots$，x_N。

$$f_\theta\left(x_0, x_1, \cdots, x_N\right)=\prod_{n=0}^{N} f_\theta\left(x_n | x_{[0, n-1]}\right), \quad (1)$$

其中 $x_{[0,n-1]} = x_{<n} = \{x_0, \cdots, x_{n-1}\}$ 为简洁起见采用的记号。在句子级别，给定前缀 p，我们将以前缀 p 为条件生成特定后缀 s 的概率记为 $f_\theta(s|p)$。

3.2. 威胁模型

在本研究中，我们聚焦于定向提取 κ 式记忆数据的威胁模型，其中我们选择 $\kappa = 1$。根据其中定义的模型知识提取，(Carlini 等人，2021年），我们假设语言模型根据最大似然准则生成一个后缀 s。随后，我们可以将定向提取正式定义为

定义 1.（定向提取）给定训练数据中包含的前缀 p 和预训练语言模型 f_θ。定向提取旨在通过$s = \underset{s'}{\mathit{argmax}}\, f_\theta(s'|p)$生成后缀。

至于 κ-情景式记忆数据，我们遵循 Carlini 等人(2021年对句子的定义：$[p, s]$ 在训练数据中至多出现 κ 个样本中。实践中，通常会对训练数据集应用截断和拼接技术，以固定生成句子的长度。如果生成句子短于指定长度，则使用填充词元补足至所需长度。本研究中，生成句子的长度为 100。

3.3. 评估指标

此前的研究使用记忆样本的数量来评估非定向提取（Carlini 等人，2021年）。为更全面地进行评估，我们使用三个指标来评估这项定向数据提取任务的表现，包括精确率 $\mathcal{M}_{\mathrm{P}}$、召回率 $\mathcal{M}_{\mathrm{R}}$ 和汉明距离 $\mathcal{M}_{\mathrm{H}}$。

精确率 $\mathcal{M}_{\mathbf{P}}$。正确生成的后缀数量占给定前缀总数的比例称为精确率 $\mathcal{M}_{\mathrm{P}}$。请注意，只有当后缀与真实值的句子长度和生成词元都完全相同时，才算正确生成。

召回率 $\mathcal{M}_{\mathbf{R}}$。正确生成的后缀数量占生成后缀总数的比例称为召回率 $\mathcal{M}_{\mathrm{R}}$。训练数据提取挑战赛中使用的指标记为 $\boldsymbol{e}_j$，其定义为：当错误生成的后缀数量达到阈值 j 时，正确恢复的后缀数量。$\boldsymbol{e}_j$ 可以评估攻击的有效性。我们定义$\mathcal{M}_{\mathrm{R}} = \frac{\boldsymbol{e}_j}{\boldsymbol{e}_{j+j}}$，在以下段落中将使用 $\mathcal{M}_{\mathrm{R}}$ 而非 e_j。在我们的实验中，j 的值与测试集大小成比例；当测试集包含 1,000 个前缀时，该值设为 100。

汉明距离 $\mathcal{M}_{\mathbf{H}}$。汉明距离表示两个等长字符串之间的差异，计算方法是统计对应符号不同的位置数。我们可以使用平均汉明距离，定量评估生成后缀与真实值之间的相似度，从而在词元级评估提取方法的性能。$\mathcal{M}_{\mathrm{H}} = \frac{1}{N}\sum_n x_n \oplus gt_n$，其中 $a \oplus b = 1$ 若 $a = b$，否则为 0。N是生成句子中的词元数量，x_n 是生成的词元，gt_n是对应的真实值词元。

我们想指出，常用的指标

用于衡量生成的句子的重复程度或质量的指标，例如齐夫系数和 REP（Welleck 等人，2019年），可能与训练数据提取的目标并不一致。因此，我们提出了上述指标，这些指标专门用于评估训练数据提取方法的有效性。这些指标 $\mathcal{M}_{\text{P}}$、$\mathcal{M}_{\text{R}}$和 $\mathcal{M}_{\text{H}}$，与提取数据的成功率密切相关，而非生成语言的质量。

4. 流程概述

基本的训练数据提取流程可以分为两个阶段。第一阶段是后缀生成，即基于一个前缀生成一组后缀。自回归语言模型 f_θ 会计算词汇表中每个词元的概率，并根据概率分布进行采样以生成下一个词元。一种用于控制生成数量的基本策略是将范围限制在排名前k 的词元内，$k = 10$这意味着语言模型只会从概率排名前 k 的词元中进行采样。第二阶段是后缀排序，即估计这些后缀的似然，并仅保留概率较高的后缀。通常，这通过基于困惑度指标的成员推断攻击来实现（Carlini 等人，2019年；2021年），具体如下

$$\mathcal{P} = \exp\left(-\frac{1}{N}\sum_{n=0}^{N}\log f_\theta(x_n|x_{[0:n-1]})\right), \qquad (2)$$

其中，N 是生成句子中的词元数量。这里的计算不包括填充词元。我们探索并评估了适用于两个阶段的各种技巧。后缀生成的结果见第5节，后缀排序的结果见第6节。

5. 改进的后缀生成

为了改进后缀生成，我们分析了真实词元和生成词元的对数几率分布。如图 2所示，这两种分布存在显著差异。为了解决这一问题，我们考察了多种自然语言处理技术的效果，例如调整采样策略和概率分布，具体如下。

5.1. 采样策略

最受欢迎的解码目标，尤其是自然语言处理任务中的条件生成，便是基于最大化的解码。它根据给定的前缀，搜索具有最高似然的后缀。由于模型是直接针对训练数据的似然进行最大化的，这种方法也适用于我们的训练数据提取任务。然而，从模型中找到理论上的 arg max 序列是不可行的（Holtzman 等人，2019年）。通常的做法是改用束搜索（Freitag &Al-Onaizan，2017年）。

表 *1.* $\mathcal{M}_{\text{P}}$、$\mathcal{M}_{\text{R}}$ 和 $\mathcal{M}_{\text{H}}$ 在束搜索中不同束数下的结果。所有结果均基于单次试验报告。

Beam	内存	$\mathcal{M}_{\text{P}}$(%) ↑	$\mathcal{M}_{\text{R}}$(%) ↑	$\mathcal{M}_{\text{H}}$ ↓
7	21531M	51.1	76.3	15.501
6	21379M	51.0	76.4	15.575
5	20731M	51.3	76.3	15.499
4	17051M	50.7	76.3	15.763
3	14333M	50.7	76.2	15.764
2	12023M	50.6	76.2	15.931
1	8783M	37.0	76.5	20.245

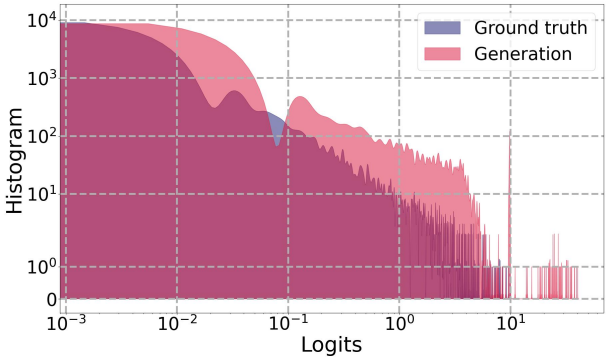

图*2*。词元 logit 直方图。该直方图展示了从 1,000 个后缀中获得的 logit 值分布，每个后缀包含 50 个词元。采用样条插值技术对直方图进行平滑处理，附录中还附有原始直方图以供参考。

作为一种近似解，束搜索只保留预先确定数量的最佳部分解。由于它在每个步长始终选择排名最靠前的部分解，因此常因缺乏多样性而受到批评（Holtzman 等人，2019年）。如表 1 所示，尽管所需内存不断增加，但当束宽度超过二时，增大束宽度所带来的性能提升会迅速减弱。由于生成过程具有随机性，采样方法始终比束搜索引入更多的多样性。因此，下面我们主要研究采样策略，包括 top-k 采样、核采样和典型采样。

Top-k 表示截断采样方法，其截断集合定义为概率最高的前 k 个词元（Fan 等人，2018年；Holtzman 等人，2018年；Radford 等人，2019年）。图3<style id='35'>（a）表明，较大的 k 可能会增加生成的句子的多样性，但也会降低精确率和汉明距离。该结果与在其他自然语言处理任务中观察到的结果一致，Meister 等人（2022年）。

Nucleus-η (Holtzman 等人，2019年) 根据累积概率截断词汇表。具体而言，选取总概率等于或大于 η 的最小高概率词元集合。 Meister 等人 (2022年) 证明，较低的 η 更有利于故事

生成。在提取任务中，最佳值则为$\eta \approx 0.6$，其提取精度比基线提高了 31%。较大或较小的 η 都会导致性能下降（图3(b)）。

典型ϕ (Meister 等人，2022年) 建议选择信息含量与预期信息含量相近的词元。具体而言，这能确保典型解码时会考虑原始分布中的概率质量。这种采样策略可以提升句子连贯性，同时减少退化重复。典型ϕ 策略等价于带有熵率约束的子集优化问题。典型ϕ 呈现出非单调趋势，这与其在抽象摘要和故事生成任务中的效果相似。

5.2. 概率分布调整

除了截断采样分布的采样策略外，本节还介绍另一种直接调整概率分布 $f_\theta(x_n|x_{<n})$ 的策略。下面我们介绍两种技巧——调整温度和重复惩罚，以优化分布。这些技巧几乎不会增加额外的计算成本，却仍能提升性能。

温度 T (Hinton 等人，2015年) 是一种带有退火项的局部重新归一化技术。温度 $T > 1$ 越高，语言模型预测的置信度越低，但也可能增加生成后缀的多样性。Carlini 等人(2021年) 开展的研究发现，在生成过程中逐渐降低温度可能会有所帮助。温度的影响见表 2。需要注意的是，随着温度升高，为了涵盖真实值所需的生成后缀数量也会增加，从而降低效率。关键在于找到多样性与效率之间的平衡。

重复惩罚是在条件语言模型（Keskar 等人，2019年）的基础上构建的。重复惩罚通过局部调整每个词元的生成概率来实现，调整依据是该词元是否重复了前一个词元。在进入 softmax 层之前，重复词元的 logit 会除以一个值 r。设置 $r > 1$会惩罚重复，而 $r < 1$ 则会鼓励重复。表3 中的结果表明，重复惩罚对训练数据提取任务的影响大多是负面的。

5.3. 曝光偏差缓解

为实现高效的向量化，在训练语言模型时，通常会将多个句子打包成一个定长序列。例如，句子“Yu 的电话号码是 12345”可能会被截断，或在训练集中加上另一个句子作为前缀，例如

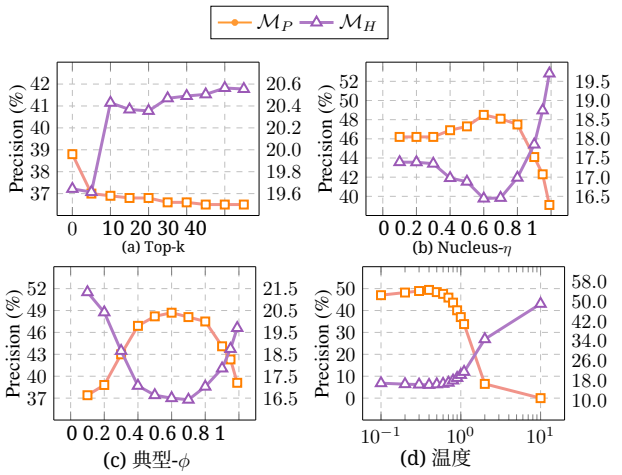

图3. 不同 top-k、nucleus-η、typical-ϕ 和温度 T取值下的实验结果。所有结果均基于5次试验报告。左侧 y 轴表示精确率 (%)(↑)，右侧表示汉明距离 (↓)。

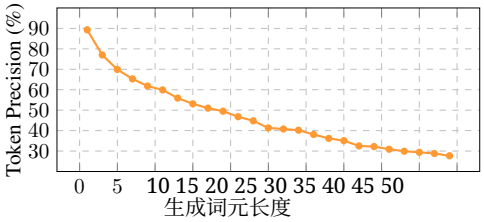

图4. 第 n 个生成词元的生成词元长度与词元精确率（%）的关系。生成后缀长度为 50。

“Yu 的电话号码是 12345”或“Yu 的地址是 XXX。Yu 的电话号码是 12345”。如表 12所示，训练集中的前缀不一定是一个完整的句子。为了更好地模拟训练设置，我们提出调整上下文窗口大小和位置偏移。

5.3.1. 动态上下文窗口

训练窗口的长度可能与提取窗口的长度不同。因此，我们提出调整上下文窗口大小，即先前生成的词元数量，如式（3）所示。此外，我们建议让不同上下文窗口大小下的结果协同确定下一个生成词元，具体如下

$$f_\theta(x_n; \mathcal{W}) = h_{\mathcal{W}}\left(f_\theta\left(x_n|x_{[n-w_1,n-1]}\right), \ldots, f_\theta\left(x_n|x_{[n-w_m,n-1]}\right)\right), \quad (3)$$

其中，$h_{\mathcal{W}}$ 表示集成方法，$\mathcal{W}$ 表示集成超参数，包括不同上下文窗口大小的数量 m 以及每个窗口大小 w_i。我们在方法中使用$m = 4$和 $w_i \in \{n, n-1, n-2, n-3\}$。精心选择超参数 m 和 w_i 或许能进一步提升性能。

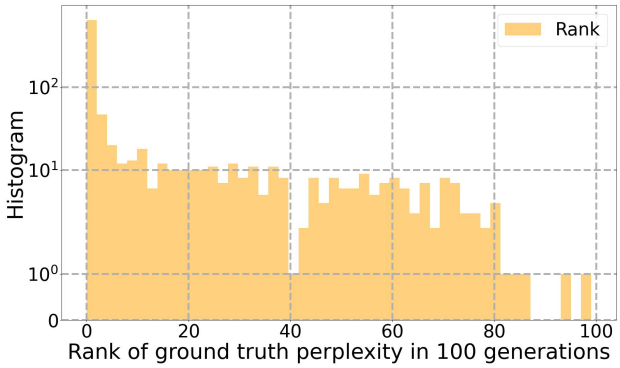

图5。 真实困惑度排名的直方图。x 轴表示真实困惑度在包含 100 个后缀困惑度的列表中的排名。

我们提出了两种实现方案。第一种方案（如公式（4）所示）是利用不同长度的先前生成词元来计算生成的概率$x_{[n-w_i,n-1]}$，然后通过对这些概率进行加权求和来得出最终概率：

$$f_\theta\left(x_n; \mathcal{W}_w\right) = \frac{\sum_{i=1}^{m} \epsilon_i f_\theta\left(x_n | x_{[n-w_i,n-1]}\right)}{\sum_{i=1}^{m} \epsilon_i}, \quad (4)$$

其中 $\mathcal{W}_w$ 表示解中的超参数，包括 m、 w_i 和 ϵ_i。 ϵ_i 表示每个概率的权重系数。式（5）中所述的第二种方案基于投票机制，其中，每个使用不同上下文窗口长度训练的模型都会为其最有把握的词元投票，形式如下：

$$f_\theta(x_n; \mathcal{W}_v) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{V}(f_\theta(x_n | x_{[n-w_i,n-1]}); \quad (5)$$

$$\mathcal{V}(f_\theta(x_n)) = \begin{cases} \rho - \mathcal{R}(f_\theta(x_n)), & \text{if } \mathcal{R}(f_\theta(x_n)) \leq \rho; \\ 0, & \text{otherwise}, \end{cases} \quad (6)$$

其中， $\mathcal{V}(\cdot)$ 表示投票函数， $\mathcal{R}(\cdot)$ 表示排名函数，它会为其有把握的词元投票。 $\mathcal{W}_v$ 表示解中的超参数，包括 w_i、 m 和 ρ。

文献Carlini 等人（2022年）指出，可提取序列的比例会随着上下文词元数量的增加呈对数线性增长。我们在实验中观察到了类似现象：随着前缀变短，词元的生成准确率会降低。然而，我们发现，结合多个上下文窗口长度可以显著提高准确率。不同窗口长度生成的概率可以进行组合，从而显著提高提取准确率。我们的实现 $\mathcal{W}_w$ 采用加权系数 $\epsilon_i = 0.9^i$，并且$\mathcal{W}_v$为其排名前 $[5, 4, 3, 2, 1]$ 5个最有把握的词元分配 $\rho = 5$ 分。表4所示结果表明，

表 2。 $\mathcal{M}_\text{P}$、 $\mathcal{M}_\text{R}$和 $\mathcal{M}_\text{H}$ 在不同温度下的结果。温度 = 1 为基线。所有结果均基于 5次试验报告。

温度	$\mathcal{M}_\text{P}$ (%)(↑)	$\mathcal{M}_\text{R}$ (%)(↑)	$\mathcal{M}_\text{H}$ (↓)
变化	48.0	76.3	19.614
0.3	48.9	76.4	16.341
1	37.0	76.5	20.245

表 3.不同重复惩罚下的 $\mathcal{M}_\text{P}$、 $\mathcal{M}_\text{R}$和 $\mathcal{M}_\text{H}$ 结果。重复惩罚 $r = 1$ 为基线。所有结果均基于5次试验报告。

重复惩罚	$\mathcal{M}_\text{P}$ (%)(↑)	$\mathcal{M}_\text{R}$ (%)(↑)	$\mathcal{M}_\text{H}$ (↓)
0.9	19.8	66.4	27.927
1	37.0	76.5	19.614
1.1	37.3	76.5	20.181
1.2	37.1	76.5	20.323
1.3	36.7	76.4	20.332
1.5	34.7	75.7	21.154

集成方法可以显著优于基线方法，准确率分别提升了143%和139%；我们还发现，加权平均策略的表现略优于投票策略。观察到的一种常见失效模式是：当生成了错误词元时，后续词元也会随之出错（示例见表12）。此处引入的窗口大小集成有助于减轻这一问题。

5.3.2. 动态位置偏移

在 GPT-Neo 等模型中，位置嵌入会添加到词元特征中。在训练期间，这一操作按句子批次进行，因此，同一个句子在不同的训练批次和生成过程中，其位置嵌入的偏移量各不相同。为了更好地提取记忆后缀，我们提出通过评估不同的偏移位置并选出“最佳”位置，来还原训练期间使用的位置。具体而言，对于给定的前缀p，我们评估不同的位置 $\mathcal{C} = c^i$，其中 c^i 是一个连续自然数列表， $c^i = \{c_1^i, \cdots\}$，使得 $|c^i| = |p|$，并计算相应的困惑度值。随后，选择困惑度最低的位置作为生成后缀的起始位置，具体如下

$$c = \underset{c^i \in \mathcal{C}}{\operatorname{argmin}} \mathcal{P}(p, c^i); \quad \hat{\phi}(x_i) = \psi(c_n) + \phi(x_n), \quad (7)$$

其中 $\psi(\cdot)$ 表示位置编码层， $\phi(\cdot)$ 表示ˆ特征映射函数，ϕ 表示由位置编码构成的特征映射函数，而 $\mathcal{P}$ 计算前缀的困惑度。实验结果见表 4。

$\mathcal{C} = \{[0, 1, \cdots, |p|], [1, 2, \cdots, |p| + 1], \cdots\}$ 进行了评估。数据表明，尽管使用位置移

表4。 $\mathcal{M}_{\text{P}}$、 $\mathcal{M}_{\text{R}}$和 $\mathcal{M}_{\text{H}}$ 在上下文窗口长度调整下的结果。所有结果均基于单次试验报告。

	$\mathcal{M}_{\text{P}}$ (%)(↑)	$\mathcal{M}_{\text{R}}$ (%)(↑)	$\mathcal{M}_{\text{H}}$ (↓)
基线	19.5	65.6	26.948
上下文窗口 $\mathcal{W}_w$	47.4	77.6	16.993
上下文窗口 $\mathcal{W}_v$	46.7	77.5	17.164
位置偏移	16.4	39.0	21.154

表5。自动调优下 $\mathcal{M}_{\text{P}}$、 $\mathcal{M}_{\text{R}}$和 $\mathcal{M}_{\text{H}}$ 的结果。所有结果均基于5次试验报告。

策略	$\mathcal{M}_{\text{P}}$ (%)(↑)	$\mathcal{M}_{\text{R}}$ (%)(↑)	$\mathcal{M}_{\text{H}}$(↓)
基线	37.0	76.5	19.614
手动选择	48.8	76.4	16.379
自动调优	49.4	76.6	16.127

位可改善 $\mathcal{M}_{\text{H}}$ 指标，但可能会对精确率和召回率产生负面影响。

5.4. 前瞻

表 12 展示了训练数据提取过程中常见的问题：只有一两个词元被错误生成或放置在不恰当的位置。为了解决这一问题，我们提出了一种技术，即向前查看 ν 步，并利用后续词元的概率来指导当前词元的生成。前瞻的目标是利用后验分布帮助计算当前词元的生成概率。我们首先给出最优概率的精确数学形式，然后介绍出于效率考虑而采用估算的实现方法。后验分布的计算方式如下：

$$f_\theta\left(x_n|x_{n+1},x_{<n}\right) = \frac{f_\theta\left(x_{n+1}|x_n,x_{<n}\right) f_\theta\left(x_n|x_{<n}\right)}{\sum_{x'_n} f_\theta\left(x_{n+1}|x'_n,x_{<n}\right) f_\theta\left(x'_n|x_{<n}\right)}. \tag{8}$$

更一般地，令**Track**(x_{start}， $x_{\text{end}}|x_{\text{cond}}$)表示以 x_{start} 开始、以 x_{end}结束且以 x_{cond}为条件的轨迹的概率乘积。那么，我们可以将 ν步后验分布写为

$$f_\theta\left(x_n|x_{n+\nu},x_{<n}\right) = \frac{\textbf{Track}(x_n,x_{n+\nu}|x_{<n})}{\sum_{x'_n}\textbf{Track}(x'_n,x_{n+\nu}|x_{<n})}, \tag{9}$$

其中**Track**的计算方式如下，

$$\begin{aligned}&\textbf{Track}(x_{\text{start}},x_{\text{end}}|x_{\text{cond}})\\ &=\sum_{x'_{\text{start}+1}}\sum_{x'_{\text{start}+2}}\cdots\sum_{x'_{\text{end}-1}} f_\theta(x_{\text{start}}|x_{\text{cond}})\\ &\qquad f_\theta(x_{\text{start}+1}|x_{\text{cond}},x_{\text{start}})\cdots\\ &\qquad f_\theta(x_{\text{end}}|x_{\text{cond}},x_{\text{start}},x'_{\text{start}+1}\ldots,x'_{\text{end}-1}).\end{aligned} \tag{10}$$

表6。 用于自动调优的超参数选择。多种最终超参数配置均可获得相当的性能，下面给出一个具有代表性的示例。

参数	范围	Step	初步	最终值
Top-k	[1, 50]	1	10	24
核心-η	[0.1, 1]	0.01	0.6	0.8
典型值-ϕ	[0.1, 1]	0.01	0.6	0.9
温度 T	[0.1, 5]	0.1	0.3	0.58
重复惩罚	[0.8, 1.3]	0.01	1	1.04

表 7。 $\mathcal{M}_{\text{P}}$、 $\mathcal{M}_{\text{R}}$和 $\mathcal{M}_{\text{H}}$ 在前瞻条件下的实验结果。所有结果均基于单次试验报告。

		$\mathcal{M}_{\text{P}}$(%)(↑)	$\mathcal{M}_{\text{R}}$(%)(↑)	$\mathcal{M}_{\text{H}}$ (↓)
基线		19.5	65.6	26.948
前瞻	λ=10	33.1	71.6	24.262
	λ=20	35.5	72.6	22.157
	λ=30	36.7	73.0	21.333

根据公式（8）中的单步前瞻公式，我们利用后验概率计算当前词元的置信度。然而，由于词汇表规模庞大（超过 50,000），我们只选择 λ 个概率为 $f_\theta($ $x'_n|x_{<n})$ 排名最高的词元，如下所示：

$$\mathcal{X} = \{x'_n|\mathcal{R}(f_\theta\left(x'_n|x_{<n}\right)) \geq \lambda\}, \tag{11}$$

其中 $\mathcal{R}(\cdot)$ 是排名函数。表 4 中详述的实验结果表明，使用前瞻策略可显著提高精确率。此外，增加 λ 的值也会相应提升性能。然而，需要注意的是，增加 λ 的值也会提高计算成本，其复杂度为 $\mathcal{O}(\lambda\cdot N)$，其中$N$表示生成词元的长度。

5.5. 超参数优化

第5.1节中提到的技巧涉及各种超参数，而直接使用最佳参数通常并非最优。手动搜索最佳超参数，也称为“人工调参”，可能会耗费大量时间。我们采用一种通用的架构自动调优方法（Akiba 等人，2019年），结合高效的搜索与剪枝策略，遵循先进框架（Snoek 等人，2012年；Koch 等人，2018年；Akiba 等人，2019年）确定优化后的超参数。我们使用协方差矩阵自适应进化策略（CMA-ES）（Hansen 等人，2003年）作为搜索算法。实验中的搜索目标设为 $\mathcal{M}_{\text{P}}$，搜索的参数包括 top-k、nucleus-η、typical-ϕ、温度 T 和重复惩罚 r。

表 8。不同排序方法下精确率、召回率和汉明距离的实验结果。所有结果均基于单次试验报告。

方法	$\mathcal{M}_{\text{P}}$(%)(↑)	$\mathcal{M}_{\text{R}}$(%)(↑)	$\mathcal{M}_{\text{H}}$(↓)
基线	37.0	76.5	19.614
困惑度 ÷Zlib	37.1	76.0	20.191
困惑度 ×Zlib	37.1	76.3	20.368
累积乘积	36.6	39.8	20.127
高置信度	40.6	77.3	19.518
意外模式	37.1	75.5	20.072

表 9。兼容性分析的精确率、召回率和汉明距离实验结果。所有结果均来自单次试验。

	$\mathcal{M}_{\text{P}}$(%) ↑	$\mathcal{M}_{\text{R}}$(%) ↑	$\mathcal{M}_{\text{H}}$ ↓
基线(GPT-Neo 1.3B)	19.5	65.6	26.948
上下文窗口 +高置信度(GPT-Neo 1.3B)	46.8	77.5	17.144
基线(GPT-Neo 2.7B)	33.5	76.6	20.921
上下文窗口 +高置信度（GPT-Neo 2.7B）	54.8	81.5	13.621

表6 列出了详细的搜索参数设置。我们还将基线参数作为搜索算法的初始值，以加快收敛。搜索轮次限制为 1,000 次。实验结果见表5。直接使用第5.1 节中列出的最佳参数，设定 k=5、 η=0.6、 ϕ=0.6、 T=0.4、r=1 ，精确率为 48.8%。实施自动调优后，结果较基线提升了 37%，且自动调优的表现略优于超参数手动调节部分。

6. 改进的后缀排序

生成多个后缀后，我们会执行排序流程，利用困惑度 $\mathcal{P}$ 作为指标，剔除可能性较低的后缀。然而，图5 中的统计分析表明，真实句子的困惑度并不总是最低。因此，我们提出了其他排序指标来解决这一差异。

6.1. 句子级标准

Zlib。 根据 Zlib压缩算法（Gailly &Adler，2004年）以比特数衡量的文本熵，是序列信息含量的定量指标。按照 Carlini 等人（2021年）的研究方法，我们将 GPT-Neo模型计算出的给定句子的困惑度与同一句子的 Zlib熵之比用作成员推断指标。此外，我们还研究了困惑度与 Zlib熵的乘积是否具有潜在效用，因为当模型对其预测表现出高度置信时，这两个指标往往都会降低。如表8所示，这两个指标对成员推断任务的整体表现仅带来了微小提升。

表10。兼容性分析的精确率、召回率和汉明距离实验结果。所有结果均基于单次试验。

	$\mathcal{M}_{\text{P}}$(%) ↑	$\mathcal{M}_{\text{R}}$(%) ↑	$\mathcal{M}_{\text{H}}$ ↓
上下文窗口 +束数=2	46.7	77.5	17.154
自动调优 +束数=2	46.4	76.2	17.331
上下文窗口 +自动调优	46.5	77.5	17.370
上下文窗口 +高置信度	46.8	77.5	17.144

累积乘积。 我们考虑了累积乘积等替代指标，它表示生成句子的联合概率，形式为 $\mathcal{L}_c = (\prod_{n=0}^{N} \log\ p(x_n|x_0, .., x_{n-1}))^{-N}$。然而，联合概率对精确率的影响微乎其微，并且会对 $\mathcal{M}_{\text{R}}$产生负面影响。

6.2. 词元级标准

奖励高置信度。 记忆数据中存在高度置信度，是被称为“记忆效应”的现象的定义性特征之一（Goodfellow 等人，2016年；Zhang 等人，2021年；Arpit 等人，2017年）。基于这一见解，我们提出采用一种策略，对包含高置信度词元的后缀给予奖励。如果句子中包含置信度较高的词元，生成词元的概率高于某一阈值，且生成词元与其他词元之间的差异也高于某一阈值，我们就提高其排名。具体而言，对于 x_n 生成后缀中的词元，如果其概率高于阈值 0.9，则从后缀 s_i 的分数中减去给定值 0.1 （ s_i 的原始分数为其困惑度）。如表 8所示，这一技巧带来了显著改进。

鼓励意外模式。 根据近期研究（Holtzman 等人，2019年），人类文本生成经常呈现这样一种模式：间歇性地纳入高困惑度词元，而不是一味选择低困惑度词元。为了解决这一问题，我们提出一种简单的方案，仅根据多数词元计算生成提示的困惑度，从而鼓励纳入意外词元（高困惑度词元）：

$$\mathcal{L}_s = \frac{1}{|\hat{\mathcal{X}}|} \sum_{x_n \in \hat{\mathcal{X}}} \log p(x_n|x_{[0:n-1]}), \tag{12}$$

$$\hat{\mathcal{X}} = \{x_j | \mu - 3\sigma < \log p(x_n|x_{[0:n-1]}) < \mu + 3\sigma\},$$

其中 μ和 σ 分别表示一个批次中$p(x_n|x_{[0:n-1]})$ 的均值和标准差。使用该方法时，在生成结果中加入意外词元不会对整体句子困惑度产生负面影响，从而提高了在成员推断中选择这些词元的似然。如表 8所示，改进相对有限。

6.3. 兼容性分析

研究发现，累积使用上述技巧并不总能带来成比例的性能提升。我们以有针对性且高效的方式研究了组合各种技术所产生的交互效应，以建立一种可行的基线方法。结果如表 10所示。应用多种上述技术并未带来显著的性能提升。即使使用束宽度 2，并保持与之前实验相同的设置，结果也没有显示出显著改进。本节所呈现的实证结果表明，要取得最佳效果，需要更加审慎且具有策略性地组合各种方法。

我们还在表格中评估了这些方法在 GPT-Neo 2.7B 上的表现 9，并得出结论：更大的预训练模型能够带来更好的结果。

7. 结论与未来工作

我们研究了十余种从语言模型中提取训练数据的技术。这些技术对生成策略和排序策略进行了细微调整。我们的实证结果表明，其中几种技巧能显著提升效果，而不同技巧之间的相互作用比预期更加微妙。此外，实证结果还表明，常用于通用文本生成的多功能方法并不总是适用于提取任务。未来的工作应着重探索开发适用于数据提取的兼容技术。可以将后缀生成和后缀排序方面的技术结合起来，并以端到端的方式进行探索。此外，我们鼓励进一步研究，以深入理解本文中某些技术表现优异背后的机制。

参考文献

Akiba, Takuya, Sano, Shotaro, Yanase, Toshihiko, Ohta, Takeru 和 Koyama, Masanori。Optuna：新一代超参数优化框架。发表于*ACM*知识发现与数据挖掘国际会议 *(KDD)*，2019年。

Arpit, Devansh, Jastrzkebski, Stanislaw, Ballas, Nicolas, Krueger, David, Bengio, Emmanuel, Kanwal, Maxin- der S., Maharaj, Tegan, Fischer, Asja, Courville, Aaron, Bengio, Yoshua 和 Lacoste-Julien, Simon。深入探究深度网络中的记忆。发表于国际机器学习会议 *(ICML)*。PMLR，2017年。

Black、Sid、Gao、Leo、Wang、Phil、Leahy、Connor 和 Biderman、Stella。《GPT-Neo：基于 Mesh-TensorFlow 的大规模自回归语言建模》，2021年。

Brown、Tom、Mann、Benjamin、Ryder、Nick、Subbiah、Melanie、Kaplan、Jared D、Dhariwal、Prafulla、Nee-lakantan、Arvind、Shyam、Pranav、Sastry、Girish、Askell、Amanda 等。《语言模型是少样本学习者》。载于《神经信息处理系统进展》（*NeurIPS*），2020年。

´Carlini, Nicholas, Liu, Chang, Erlingsson, Ulfar, Kos, Jernej 和 Song, Dawn。秘密共享者：评估和测试神经网络中的非预期记忆。发表于*USENIX*安全研讨会论文集 *(SEC)*，2019年。

Carlini, Nicholas、Tramer, Florian、Wallace, Eric、Jagielski, Matthew、Herbert-Voss, Ariel、Lee, Katherine、Roberts, Adam、Brown, Tom、Song, Dawn、Erlingsson, Ulfar 等。提取大语言模型的训练数据。载于*USENIX*安全研讨会论文集 *(SEC)*，2021年。

Carlini, Nicholas、Ippolito, Daphne、Jagielski, Matthew、Lee, Katherine、Tramer, Florian 和 Zhang, Chiyuan。量化神经语言模型中的记忆。*arXiv*预印本 *arXiv:2202.07646*，2022年。

Choquette-Choo, Christopher A、Tramer, Florian、Carlini, Nicholas 和 Papernot, Nicolas。仅基于标签的成员推断攻击。载于国际机器学习会议 *(ICML)*。PMLR，2021年。

Fan, Angela、Lewis, Mike 和 Dauphin, Yann。分层神经故事生成。载于计算语言学协会年会论文集 *(ACL)*，2018年。

Feldman, Vitaly. 学习是否需要记忆？一个关于长尾的简短故事。载于*ACM SIGACT*计算理论研讨会（*STOC*）论文集，2020年。

Fredrikson, Matt, Jha, Somesh, and Ristenpart, Thomas. 利用置信度信息的模型反演攻击及基本对策。载于*ACM SIGSAC*计算机与通信安全会议（*ACM CCS*）论文集，2015年。

Freitag, Markus 和 Al-Onaizan, Yaser。神经机器翻译的束搜索策略。载于首届神经机器翻译研讨会论文集，第56–60页，温哥华，2017年8月。计算语言学协会。数字对象标识符：10.18653/v1/W17-3207。统一资源定位符https://aclanthology.org/W17-3207。

Gailly, Jean-loup 和 Adler, Mark。《Zlib压缩库》，2004年。

Ganju, Karan、Wang, Qi、Yang, Wei、Gunter, Carl A 和 Borisov, Nikita。利用置换不变表示对全连接神经网络进行属性推断攻击。载于*ACM SIGSAC*计算机与通信安全会议论文集（*ACM CCS*），2018年。

Gao、Leo、Biderman、Stella、Black、Sid、Golding、Laurence、Hoppe、Travis、Foster、Charles、Phang、Jason、He、Horace、Thite、Anish、Nabeshima、Noa 等。《The Pile：用于语言建模的多样化文本 800GB数据集》。*arXiv* 预印本 *arXiv:2101.00027*，2020年。

Gong, Neil Zhenqiang 和 Liu, Bin。你由你认识的人和你的行为定义：通过用户的社交好友和行为发起属性推断攻击。载于 *USENIX* 安全研讨会（*SEC*）论文集，2016年。Gong, Xueluan、Wang, Qian、Chen, Yanjiao、Yang, Wang 和 Jiang, Xinchang。针对基于云的机器学习模型的模型提取攻击与防御。*IEEE*通信杂志，58(12)，2020年。Goodfellow, Ian、Bengio, Yoshua 和 Courville, Aaron。深度学习。MIT出版社，2016年。Hansen, Nikolaus, M¨uller, Sibylle D 和 Koumoutsakos, Petros。通过协方差矩阵自适应（cma-es）降低去随机化进化策略的时间复杂度。《进化计算》，11(1)，2003年。Hayes, J、Melis, L、Danezis, G 和 De Cristofaro, E。Logan：针对生成模型的成员推断攻击。载于 隐私增强技术论文集（*PoPETs*）。德格鲁伊特出版社，2019年。He, Yang、Rahimian, Shadi、Schiele, Bernt 和 Fritz, Mario。Segmentations-leak：语义图像分割中的成员推断攻击与防御。载于 欧洲计算机视觉会议（*ECCV*）。Springer，2020年。Hilprecht, Benjamin, H¨arterich, Martin 和 Bernau, Daniel。针对生成模型的蒙特卡罗攻击与重建式成员推断攻击。隐私增强技术论文集，2019(4)，2019 年。Hinton, Geoffrey E.、Vinyals, Oriol 和 Dean, Jeffrey。提炼神经网络中的知识。 *CoRR*，abs/1503.02531，2015年。统一资源定位符 http://arxiv.org/abs/1503.02531。

Holtzman, Ari、Buys, Jan、Forbes, Maxwell、Bosselut, An-toine、Golub, David 和 Choi, Yejin。借助协作式判别器学习写作。载于 计算语言学协会年会论文集（*ACL*），2018年。

Holtzman, Ari、Buys, Jan、Du, Li、Forbes, Maxwell 和 Choi, Yejin。神经文本退化的奇特案例。载于 国际学习表征会议（*ICLR*），2019年。

Hu、Hongsheng、Salcic、Zoran、Sun、Lichao、Dobbie、Gillian、Yu、Philip S 和 Zhang、Xuyun。《机器学习中的成员推断攻击：综述》。*ACM*计算综述（*CSUR*），54(11s)，2022年。

Jagielski、Matthew、Thakkar、Om、Tramer、Florian、Ippolito、Daphne、Lee、Katherine、Carlini、Nicholas、Wallace、Eric、Song、Shuang、Thakurta、Abhradeep、Papernot、Nicolas 等。测量对记忆化训练示例的遗忘。*arXiv* 预印本 *arXiv:2207.00099*，2022年。

Jayaraman, Bargav、Ghosh, Esha、Inan, Huseyin、Chase, Melissa、Roy, Sambuddha 和 Dai, Wei。针对生成式语言模型的主动数据模式提取攻击。*arXiv* 预印本 *arXiv:2207.10802*，2022年。Juuti, Mika、Szyller, Sebastian、Marchal, Samuel 和 Asokan, N。Prada：防御深度神经网络模型窃取攻击。收录于*IEEE*欧洲安全与隐私研讨会（*EuroS&P*）。IEEE，2019年。Kandpal, Nikhil、Wallace, Eric 和 Raffel, Colin。对训练数据进行去重可降低语言模型的隐私风险。*arXiv* 预印本 *arXiv:2202.06539*，2022年。Kaya, Yigitcan 和 Dumitras, Tudor。数据增强何时有助于抵御成员推断攻击？收录于国际机器学习会议（*ICML*）。PMLR，2021年。Kenton, Jacob Devlin、Ming-Wei Chang 和 Toutanova, Lee Kristina。BERT：面向语言理解的深度双向变换器预训练。收录于北美计算语言学协会分会会议（*NAACL*），2019年。Keskar, Nitish Shirish、McCann, Bryan、Varshney, Lav R、Xiong, Caiming 和 Socher, Richard。CTRL：用于可控生成的条件变换器语言模型。*arXiv* 预印本 *arXiv:1909.05858*，2019年。Koch, Patrick、Golovidov, Oleg、Gardner, Steven、Wujek, Brett、Griffin, Joshua 和 Xu, Yan。AutoTune：用于超参数调优的无导数优化框架。收录于*ACM*国际知识发现与数据挖掘会议（*KDD*），2018年。Lehman, Eric、Jain, Sarthak、Pichotta, Karl、Goldberg, Yoav 和 Wallace, Byron C。使用临床笔记预训练的 BERT 是否会泄露敏感数据？收录于北美计算语言学协会分会会议（*NAACL*），2021年。

Leino, Klas 和 Fredrikson, Matt。被盗的记忆：利用模型记忆实现经过校准的白盒成员推断。收录于*USENIX*安全专题讨论会论文集（*SEC*），2020年。

Liu, Jinghui, Capurro, Daniel, Nguyen, Anthony, 和 Verspoor, Karin。“注释膨胀”会影响用于临床预测任务的基于深度学习的自然语言处理模型。《生物医学信息学杂志》（*JBI*），133，2022年。

Meister, Clara, Pimentel, Tiago, Wiher, Gian, 和 Cotterell, Ryan。局部典型采样。计算语言学协会汇刊（*TACL*），2022年。

Naseri, Mohammad、Hayes, Jamie 和 De Cristofaro, Emil- iano。迈向联邦学习的鲁棒性与隐私：对本地差分隐私和中心化差分隐私的实验。*arXiv* 电子预印本，2020年。Parisot, M、Spagnuelo, D 等。针对卷积神经网络的属性推断攻击：目标模型复杂度的影响及其含义。《国际安全与密码学会议论文集》，2021年。Radford, Alec、Wu, Jeffrey、Child, Rewon、Luan, David、Amodei, Dario、Sutskever, Ilya 等。语言模型是无监督多任务学习器。*OpenAI* 博客，1(8)，2019年。Rezaei, Shahbaz 和 Liu, Xin。成员推断攻击的难点。载于《*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*）》，2021年。Sablayrolles, Alexandre、Douze, Matthijs、Schmid, Cordelia、Ollivier, Yann 和 J´egou, Herv´e。白盒与黑盒：成员推断的贝叶斯最优策略。载于《国际机器学习会议（*ICML*）》。PMLR，2019年。Salem, Ahmed、Zhang, Yang、Humbert, Mathias、Berrang, Pascal、Fritz, Mario 和 Backes, Michael。ML-Leaks：针对机器学习模型的模型与数据无关的成员推断攻击及防御。*arXiv* 预印本 *arXiv:1806.01246*，2018年。Shokri, Reza、Stronati, Marco、Song, Congzheng 和 Shmatikov, Vitaly。针对机器学习模型的成员推断攻击。载于*IEEE*安全与隐私研讨会（*SP*）。IEEE，2017年。Snoek, Jasper、Larochelle, Hugo 和 Adams, Ryan P。机器学习算法的实用贝叶斯优化。载于《神经信息处理系统进展》（*NeurIPS*），第25卷，2012年。

Song, Liwei 和 Mittal, Prateek。机器学习模型隐私风险的系统评估。载于《*USENIX* 安全研讨会》（*SEC*）论文集，2021年。

Thomas, Aleena, Adelani, David Ifeoluwa, Davody, Ali, Mogadala, Aditya, 和 Klakow, Dietrich。研究预训练词嵌入对神经网络记忆能力的影响。载于国际文本、语音与对话会议（*TSD*）。Springer，2020年。

Tople, Shruti, Sharma, Amit, 和 Nori, Aditya。通过因果学习缓解隐私攻击。载于国际机器学习会议（*ICML*）。PMLR，2020年。

Tram`er, Florian, Zhang, Fan, Juels, Ari, Reiter, Michael K, 和 Ristenpart, Thomas. 通过预测 {API} 窃取机器学习模型。载于*USENIX* 安全研讨会（*SEC*）论文集，2016年。Wang, Yijue, Wang, Chenghong, Wang, Zigeng, Zhou, Shanglin, Liu, Hang, Bi, Jinbo, Ding, Caiwen, 和 Ra-jasekaran, Sanguthevar. 应对成员推断攻击：剪枝便已足够。载于国际人工智能联合会议（*IJCAI*），2021年。Welleck, Sean, Kulikov, Ilia, Roller, Stephen, Dinan, Emily, Cho, Kyunghyun, 和 Weston, Jason. 通过非似然训练进行神经文本生成。载于国际学习表征会议（*ICLR*），2019年。Wolf, Thomas, Debut, Lysandre, Sanh, Victor, Chaumond, Julien, Delangue, Clement, Moi, Anthony, Cistac, Pierric, Rault, Tim, Louf, R´emi, Funtowicz, Morgan, 等。Transformer：最先进的自然语言处理技术。载于自然语言处理实证方法会议论文集（*EMNLP*），2020年。Wu, Bang, Yang, Xiang wen, Pan, Shirui, 和 Yuan, Xingliang. 图神经网络的模型提取攻击：分类体系与实现。载于*ACM* 亚洲计算机与通信安全会议论文集（*ACM ASIA-CCS*），2022年。Yeom, Samuel, Giacomelli, Irene, Fredrikson, Matt, 和 Jha, Somesh. 机器学习中的隐私风险：分析其与过拟合的关联。载于*IEEE* 计算机安全基础研讨会（*CSF*）。IEEE，2018年。Zhang, Chiyuan, Bengio, Samy, Hardt, Moritz, Recht, Ben-jamin, 和 Vinyals, Oriol. 理解深度学习（仍然）需要重新思考泛化。*ACM* 通讯，64(3)，2021年。

A. 附录

A.1. 指南

近年来，提取语言模型训练数据的重要性日益受到重视。通过考察本文中的各种技术，我们发现，某些技巧的效果不如基线方法，或者会带来巨大的计算开销。因此，我们提出以下指南，以指导语言模型数据提取方法的开发。

1. 可用性：要使数据提取方法被视为实用方法，通常需要其采用门槛低且易于使用。例如，不需要额外数据集，实现方式简单，无需复杂模块，并且能够轻松适配各种应用。

2. 有效性：预期数据提取方法在给定数据集上展现出良好的性能 s.

3. 效率：这些方法必须具备较高的计算效率，并充分利用有限的计算成本。此外，最好采用小而有效的模块。额外的语言模型可以提升性能，但我们期望这些方法能够用小型模块替代它们，且不会明显牺牲性能。

4. 有害性。作为一种针对数据保护的对抗性任务，我们期望语言模型的数据提取方法具有危害性，因为强有力的有害提取攻击能够更准确地评估信息泄露风险，相当于为潜在泄露风险提供更高的下界。

我们想强调，这些指南旨在提高训练数据提取方法的成功率，从而更准确地评估语言模型的隐私信息泄露，而非提出有害的攻击方法。

A.2. 训练数据提取的典型失败案例

本节中，我们观察训练数据提取过程中出现的经典失败实例，并将其分为两类，如表12所示。第一类失败是生成后缀中间部分的词元有误，导致后续词元生成不准确。第二类失败是后缀中只有少量词元与真实情况不符，而后续词元生成正确。

前瞻方法的提出，源于训练数据提取过程中遇到的第二类失败。此外，从数据提取成功和失败案例的示例中可以明显看出，前缀和后缀都不是完整的句子。例如，前缀可能以标点符号“.”开头，也可能是像“任何形式的保证”这样的残句。这是训练语言材料中普遍采用截断和填充技术所导致的。鉴于这一现象，本文提出了动态上下文窗口和动态位置偏移方法。

与失败案例相比，表 11 列举了一些成功提取的示例。这些示例有助于直观地理解训练数据提取任务。

A.3. 未经插值的直方图

A.3.1. 句子困惑度直方图

为了考察生成后缀和真实后缀的句子困惑度分布，我们使用直方图直观呈现它们之间的差异。我们计算真实后缀和多个生成后缀的句子困惑度，然后计算真实困惑度的排名，并据此在图 5中绘制直方图，其中横轴的计算方式为

$$\mathcal{S}(\mathcal{P}_{gt}, \{\mathcal{P}_{s_i}\}), \tag{13}$$

其中， $\mathcal{S}(a,\ b)$ 返回 a 在集合 b 中的排名， $\mathcal{P}_{gt}$ 是给定前缀 p 的真实后缀的困惑度， s_i 表示给定前缀的第 i 个生成后缀 p。

A.3.2. 标记对数几率直方图

为了研究生成词元与真实词元之间对数几率值的分布，我们使用直方图来直观展示它们的差异。图中展示了真实后缀和生成后缀的词元对数几率直方图 6，可以观察到对数几率分布存在明显差异。此外，为了深入了解不同位置上对数几率值的分布，图中还给出了各个位置的对数几率值直方图7、8和 9。

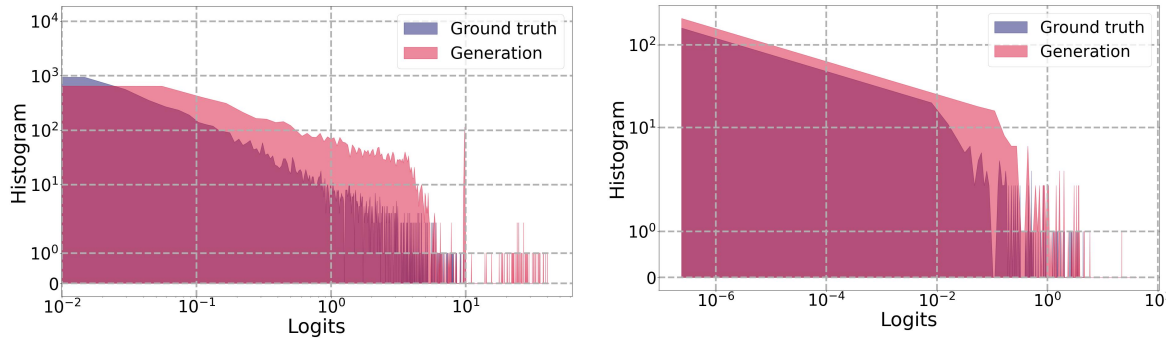

图*6.* 词元 logit 直方图。该直方图展示图*7.* 第 0 个（第一个）生成词元的对数几率值。该直方图展示了从 1,000 个后缀中获得的 logit 值分布，每个后缀包含 50 个词元。第 0 个词元的对数几率值分布来自 1,000 个后缀。

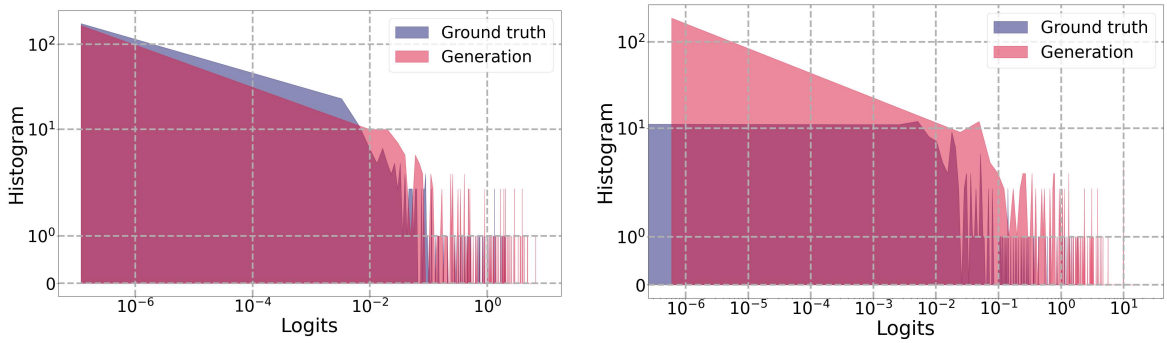

图*8.* 第 24 个（中间）生成词元的对数几率值。图*9.* 第 49 个（最后一个）生成词元的对数几率值。该直方图展示了从 1,000 个后缀的第 24 个词元获得的对数几率值分布。该直方图展示了从 1,000 个后缀的第 49 个词元获得的对数几率值分布。

A.4. 要点

我们总结出以下几点：

1. 在训练数据提取的两个阶段（后缀生成和后缀排序）中，后缀生成对提取精度的影响比后缀排序更大。

2. 动态上下文窗口是比其他技巧实用得多的一种方法。

3. 采样策略对超参数非常敏感。

4. 我们建议使用不超过两束，因为束数增加会显著提高内存需求，但精确率的提升幅度会逐渐减小。

5. 前瞻机制提高了当前生成词元的精确率，提升程度取决于所考虑的未来词元数量。

表 *11*。训练数据　　提取成功案例。生成后缀与真实后缀的长度和令牌数均相同　　ns.

前缀	/* * 版权所有 2003年-2006年 Sun Microsystems, Inc.。保留所有权利。 * 请勿更改或删除版权声明或此文件头。 * * 此代码是自由软件;
生成后缀	适销性或 * 特定用途适用性。请参阅 GNU通用公共许可证 * 第 2 版以了解更多详情（LICENSE文件中附有一份副本， * 随附于此代码）。
前缀	特此免费授予任何获得本软件及相关 文档文件（“软件”）副本的人不受限制地处理本软件，包括 但不限于以下权利:
生成后缀	包括但不限于适销性保证　　, 特定用途适用性 和不侵权保证。在任何情况下，作者或 版权所有者均不对.....承担责任

A.5. 负面社会影响

本研究探讨了提取训练数据的各种技术。必须注意的是，这些方法可能会被用于不良目的。本文聚焦于一种特定类型的训练数据提取，即目标训练数据提取，这种提取需要训练数据集中存在一个前缀。如果攻击者能够访问语言模型的训练数据，或能够生成与训练数据相似的句子，就有可能恢复训练数据中的句子，如表11所示。

在这项工作中，我们着力减轻负面伦理影响，重点研究特定的语言模型 GPT-Neo，该模型使用 Pile 作为训练数据集。请注意，Pile 本身已是一个公开数据集，其数据主要来自学术或专业来源，而非私人来源。

训练数据提取的主要预期用户包括旨在调查语言模型隐私与安全特征的机构和组织。然而，必须承认，这些方法也可能被恶意攻击者利用。短期来看，一些语言模型存在信息泄露风险。从长远来看，我们呼吁语言学界采取措施降低这一风险，并开发能够抵御包括成员推断攻击、训练数据提取攻击等攻击的安全语言模型。

表12。训练数据提取示例。红色词元表示生成后缀与真实后缀之间不匹配的词元。

前缀	.
	//
	// 版权所有 (c) 2008-2011 德州仪器公司。保留所有权利。
	// 软件许可协议
	//
	// 德州仪器 (TI) 提供此软件，仅供
	// 仅限于 TI

生成后缀	不限于 , 适销性和特定用途的默示担保适用于
	// 本软件。
	// 源代码和二进制形式的再分发和使用，须符合

真实值后缀	不仅限于 , 适销性默示担保及特定用途适用性
	// 本软件适用特定用途适用性保证。TI 不得 , 在任何
	// 情形 , 都须

前缀	. 暂无评论： 朋友 不满者 《伟大的变革》随灵感而发，不定期出版。本网站上的文字纯属个人观点 ，仅代表 Albert Bates，并遵循知识共享署名非
生成后缀	（非商业）：不得将本作品用于商业用途。未经许可不得用于商业用途 版权所有者的明确书面同意。
真实值后缀	(非商业)：您不得将本作品用于商业目的。 相同方式共享 (相同方式共享)：如果您修改、转换或 基于本作品进行创作，您只能依据与本许可相同或相似的许可分发由此产生的作品 许可。任何内容均不
前缀	# 特此免费授予任何人 # 获取本软件及相关文档的副本 # 文件（以下简称“软件”），并可不受限制地使用软件 # ,
生成后缀	任何形式的保证 , # 明示或默示 , 包括但不限于 的 # 适销性 , 特定用途适用性及 # 不侵权
真实值后缀	任何形式的保证 , # 明示或默示 , 包括但不限于保证 # 关于 适销性 , 特定用途适用性及 # 不侵权
前缀	//// //// 本软件按“原样”提供，且不提供任何 //// //// 明示或默示保证 , 包括
生成后缀	损害赔偿 //// ////（包括但不限于采购替代商品//// ////或服务；使用损失、数据或利润；或业务
真实值后缀	损害赔偿 //// ////（包括但不限于采购替代品 //// //// 商品 或服务；使用损失、数据或利润；